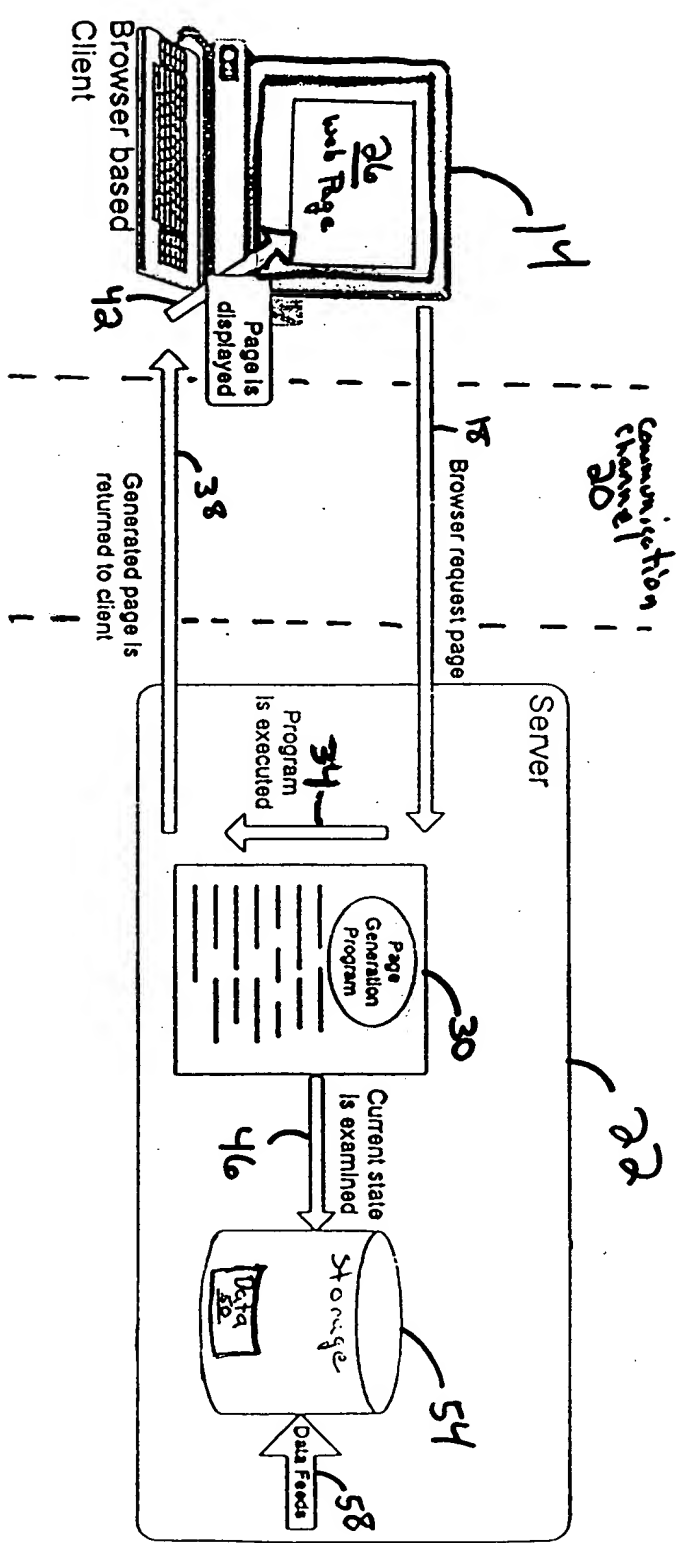


FIG. 1 is a block diagram of a system for generating a page.



10

FIG. 1
Prior Art

200

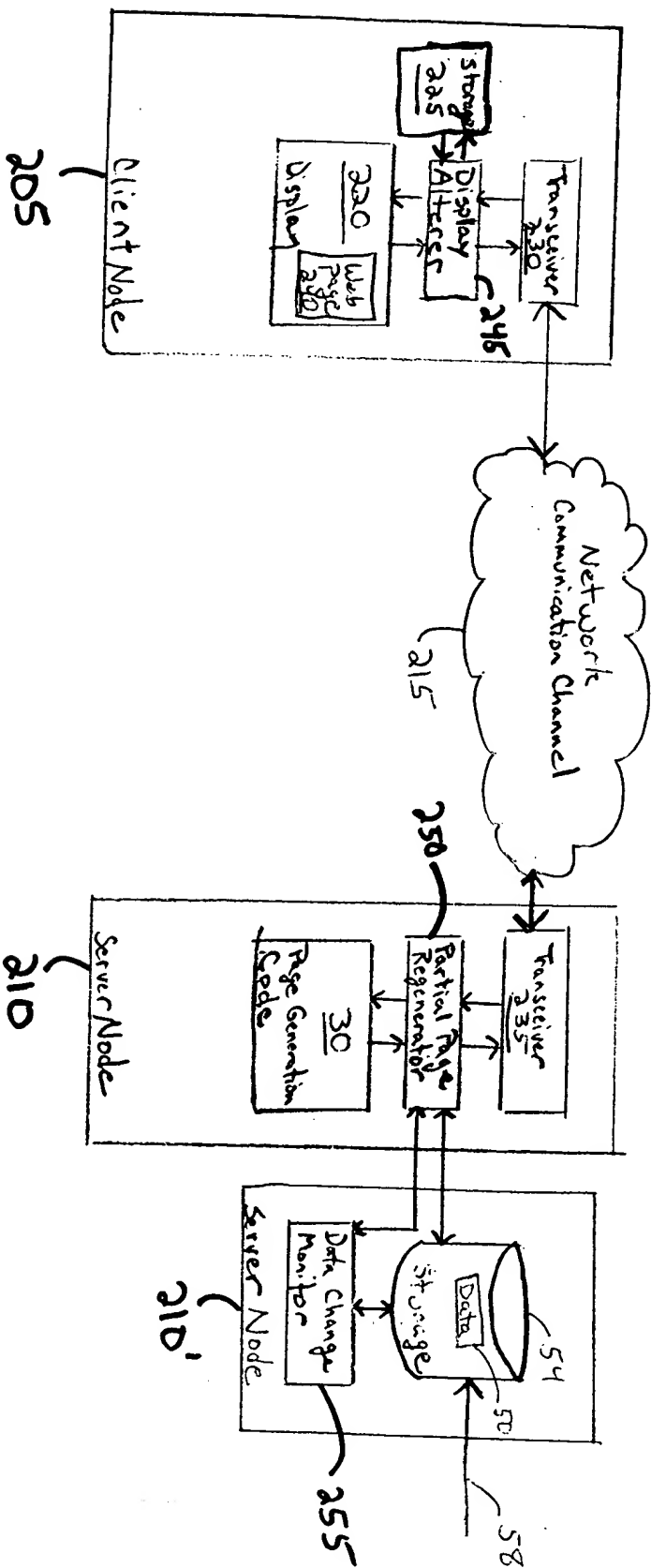


FIG. 2a

FIG. 2a

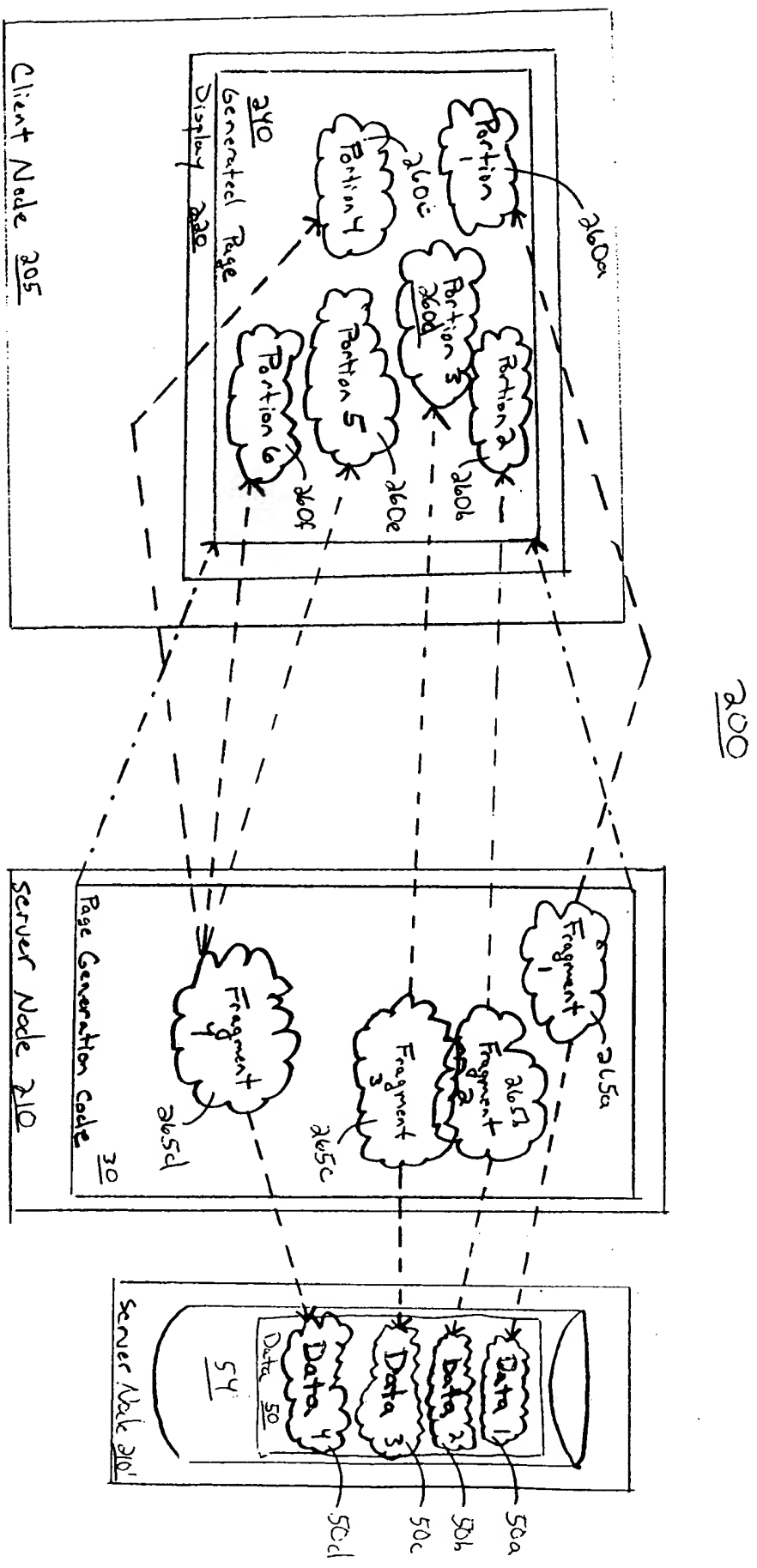


FIG. 2b

FIG. 3a is a block diagram of a system 300 for processing a request to access a resource. The system 300 includes a client 310, a server 320, and a database 330. The client 310 sends a request to the server 320, which then accesses the database 330 to retrieve the requested resource. The system 300 is configured to handle requests for resources stored in the database 330.

Original Servlet Code

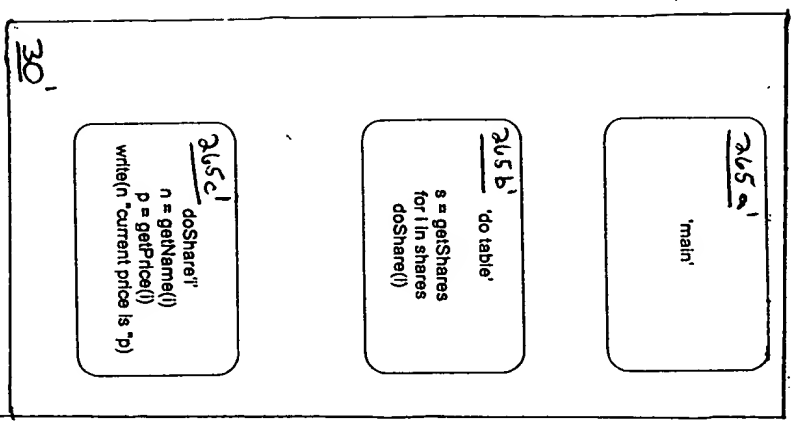


FIG. 3a

Wrapped Servlet Code

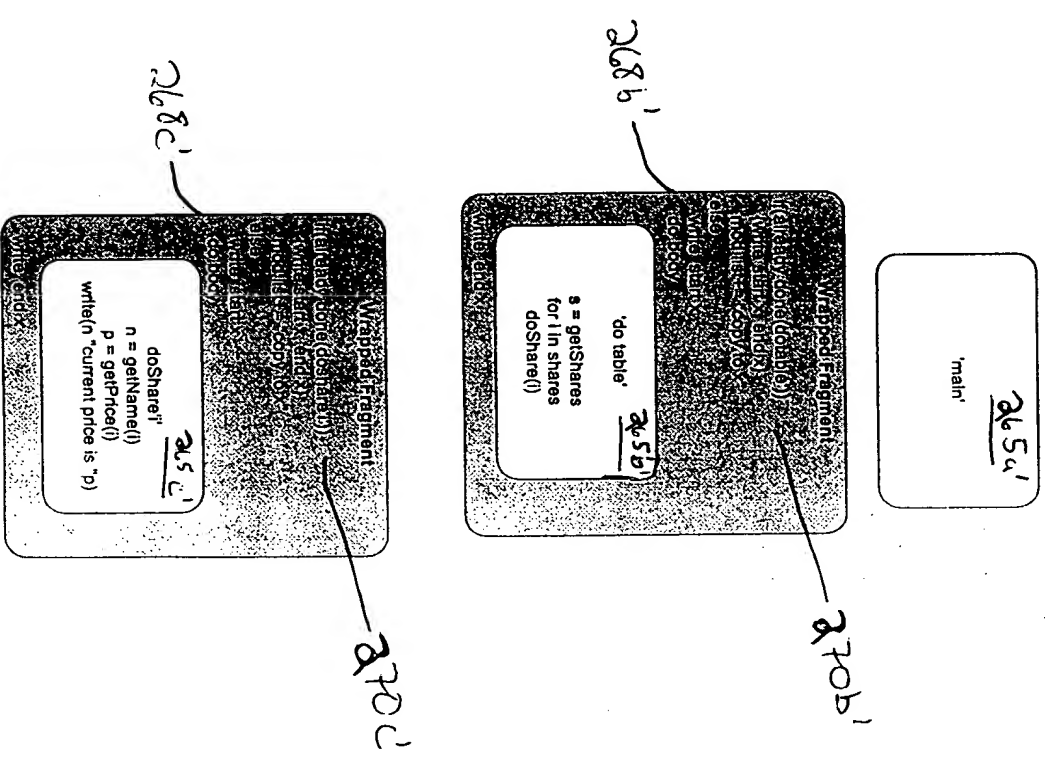


FIG. 3b

FIG. 4a

Original CGI Code

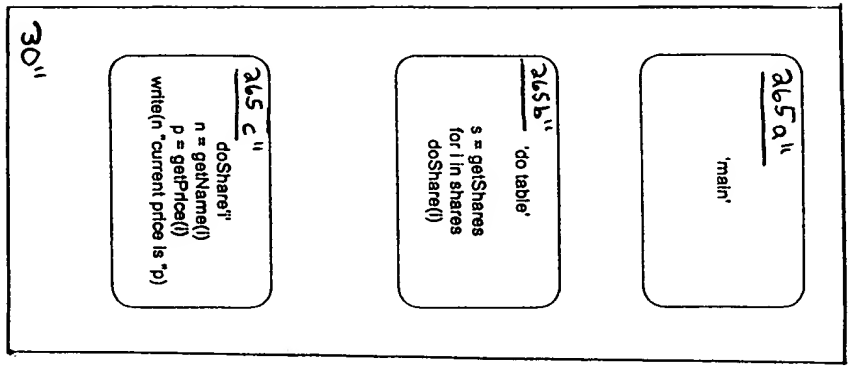


FIG. 4a

Runtime wrapping by Intercepting CGI Interpreter

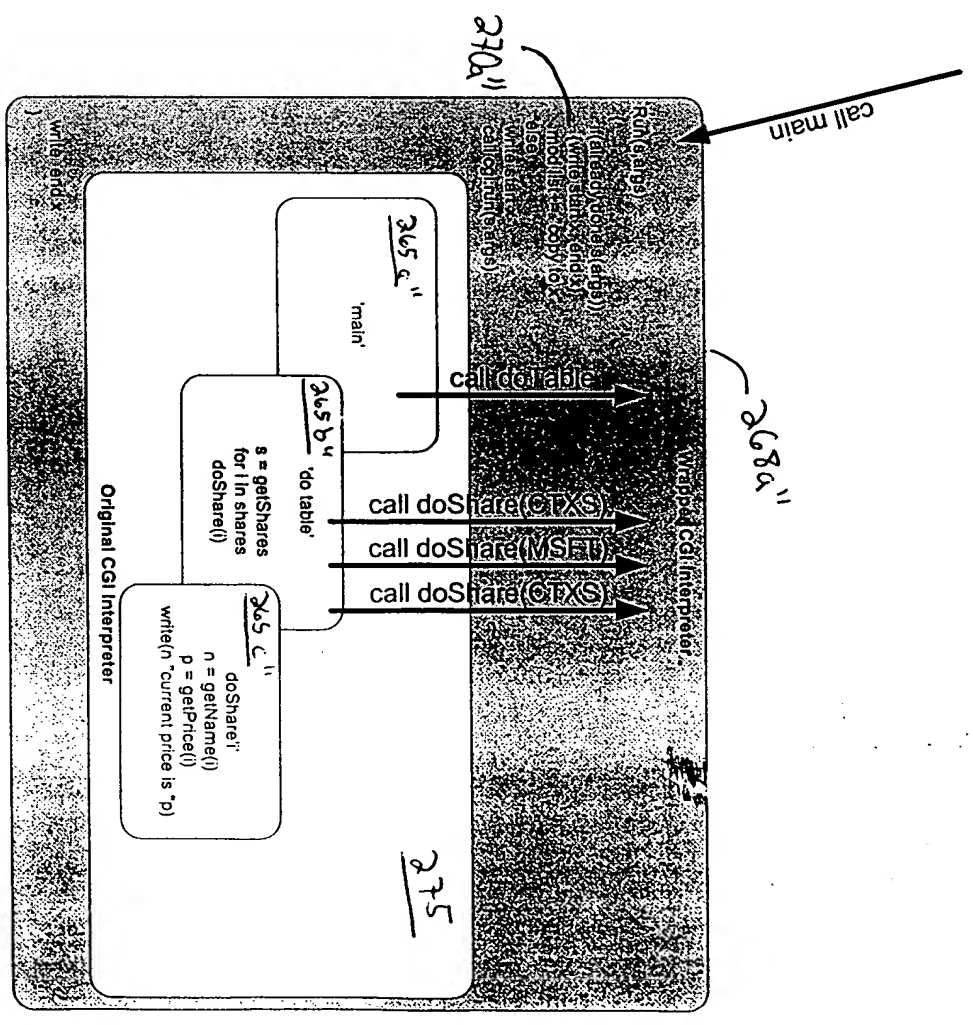


FIG. 4b

FIG. 4c

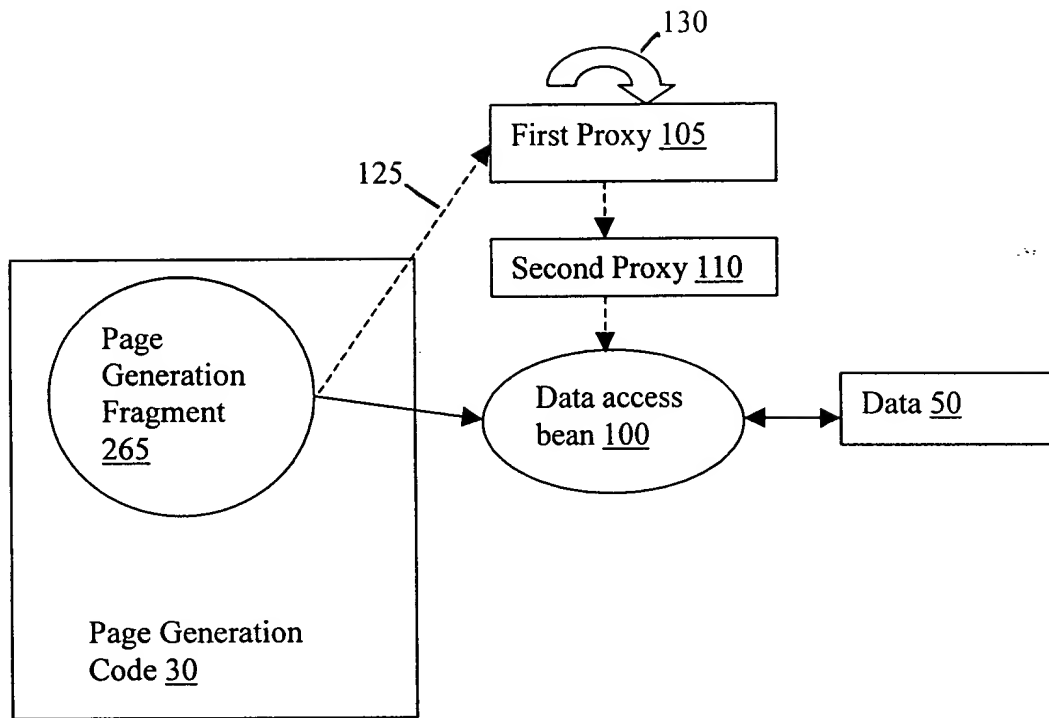


FIG. 5a is a block diagram of a system architecture. The system includes a main module (265a) which is connected to a series of modules (270a, 265b, 270b, 265c, 270c, 265d, 270d, 265e, 270e, 265f, 270f, 265g, 270g, 265h, 270h, 265i, 270i, 265j, 270j, 265k, 270k, 265l, 270l, 265m, 270m, 265n, 270n, 265o, 270o, 265p, 270p, 265q, 270q, 265r, 270r, 265s, 270s, 265t, 270t, 265u, 270u, 265v, 270v, 265w, 270w, 265x, 270x, 265y, 270y, 265z, 270z). The main module (265a) is connected to module 270a, which is connected to module 265b, which is connected to module 270b, which is connected to module 265c, which is connected to module 270c, which is connected to module 265d, which is connected to module 270d, which is connected to module 265e, which is connected to module 270e, which is connected to module 265f, which is connected to module 270f, which is connected to module 265g, which is connected to module 270g, which is connected to module 265h, which is connected to module 270h, which is connected to module 265i, which is connected to module 270i, which is connected to module 265j, which is connected to module 270j, which is connected to module 265k, which is connected to module 270k, which is connected to module 265l, which is connected to module 270l, which is connected to module 265m, which is connected to module 270m, which is connected to module 265n, which is connected to module 270n, which is connected to module 265o, which is connected to module 270o, which is connected to module 265p, which is connected to module 270p, which is connected to module 265q, which is connected to module 270q, which is connected to module 265r, which is connected to module 270r, which is connected to module 265s, which is connected to module 270s, which is connected to module 265t, which is connected to module 270t, which is connected to module 265u, which is connected to module 270u, which is connected to module 265v, which is connected to module 270v, which is connected to module 265w, which is connected to module 270w, which is connected to module 265x, which is connected to module 270x, which is connected to module 265y, which is connected to module 270y, which is connected to module 265z, which is connected to module 270z.

| Output | | ModList | |
|--------|---------------------|---------|-------------|
| a | | | |
| a | | 290a | assign 1->0 |
| b | <table> | | 285a |
| c | <tr> | | |
| c | <td>Name</td> | | |
| c | <td>Price </td> | | |
| c | </tr> | 290b | |
| d | | | |
| e | <tr> | | |
| e | <td>ABC Corp. </td> | | |
| e | <td>99.9</td> | 295b | |
| e | </tr> | | |
| d | | | |
| f | | | |
| g | <tr> | 290c | |
| g | <td>XYZ Corp. </td> | | |
| g | <td>1.2</td> | | |
| g | </tr> | 295c | |
| f | | | |
| h | | 290d | copy 2->4 |
| h | | | |
| c | </table> | 295d | |
| b | | | |
| a | | 295a | |

FIG. 5b

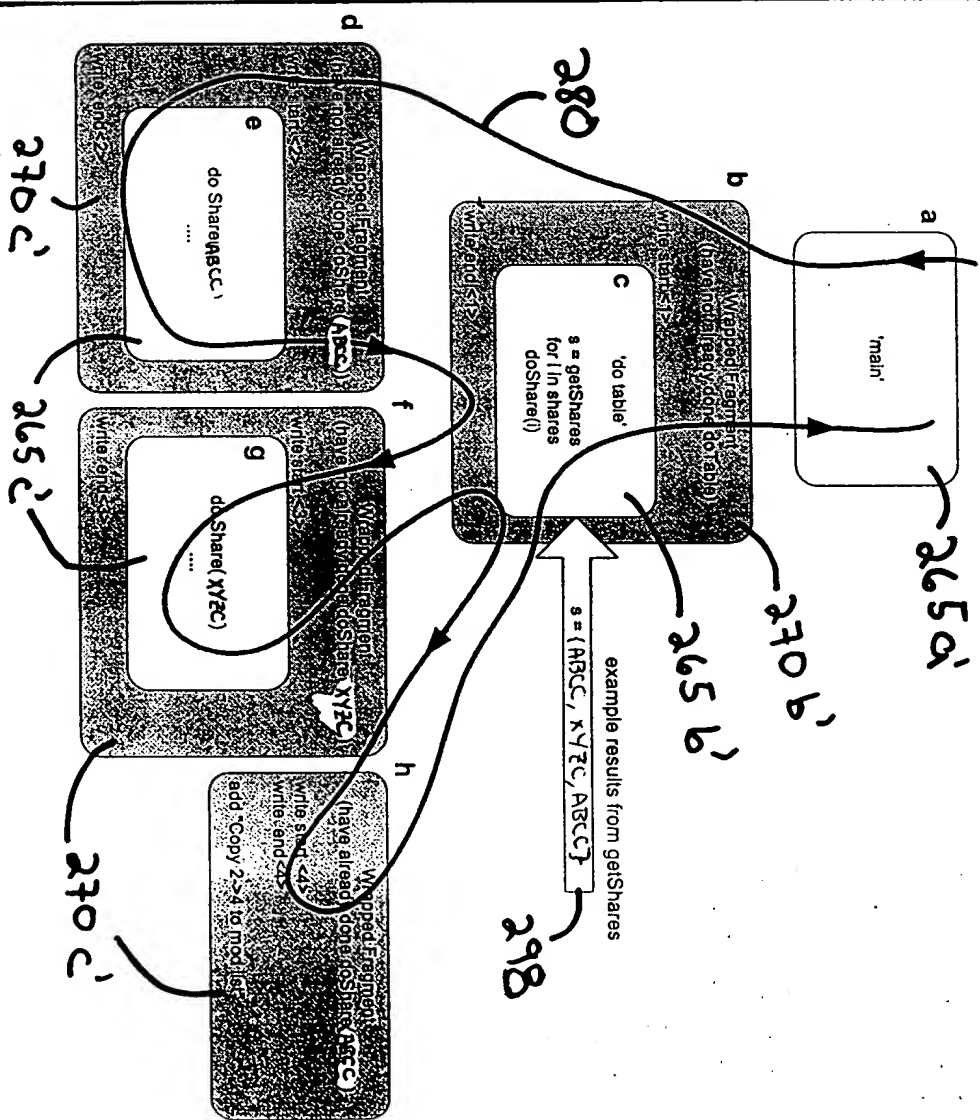


FIG. 5a



FIG. 6

FIG. 1 is a block diagram of a system for generating a value from a key.

300

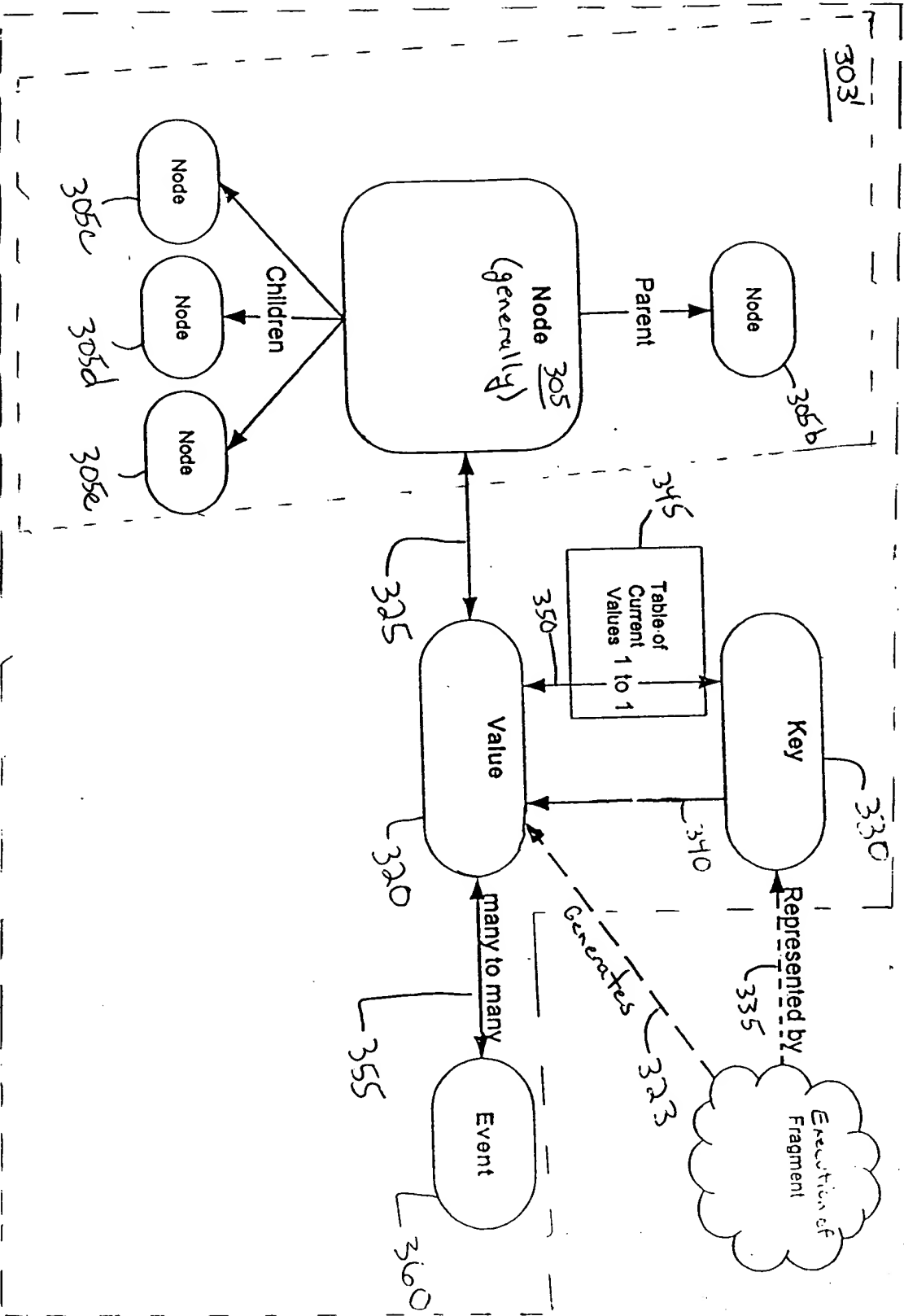
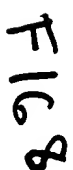


FIG 7

308

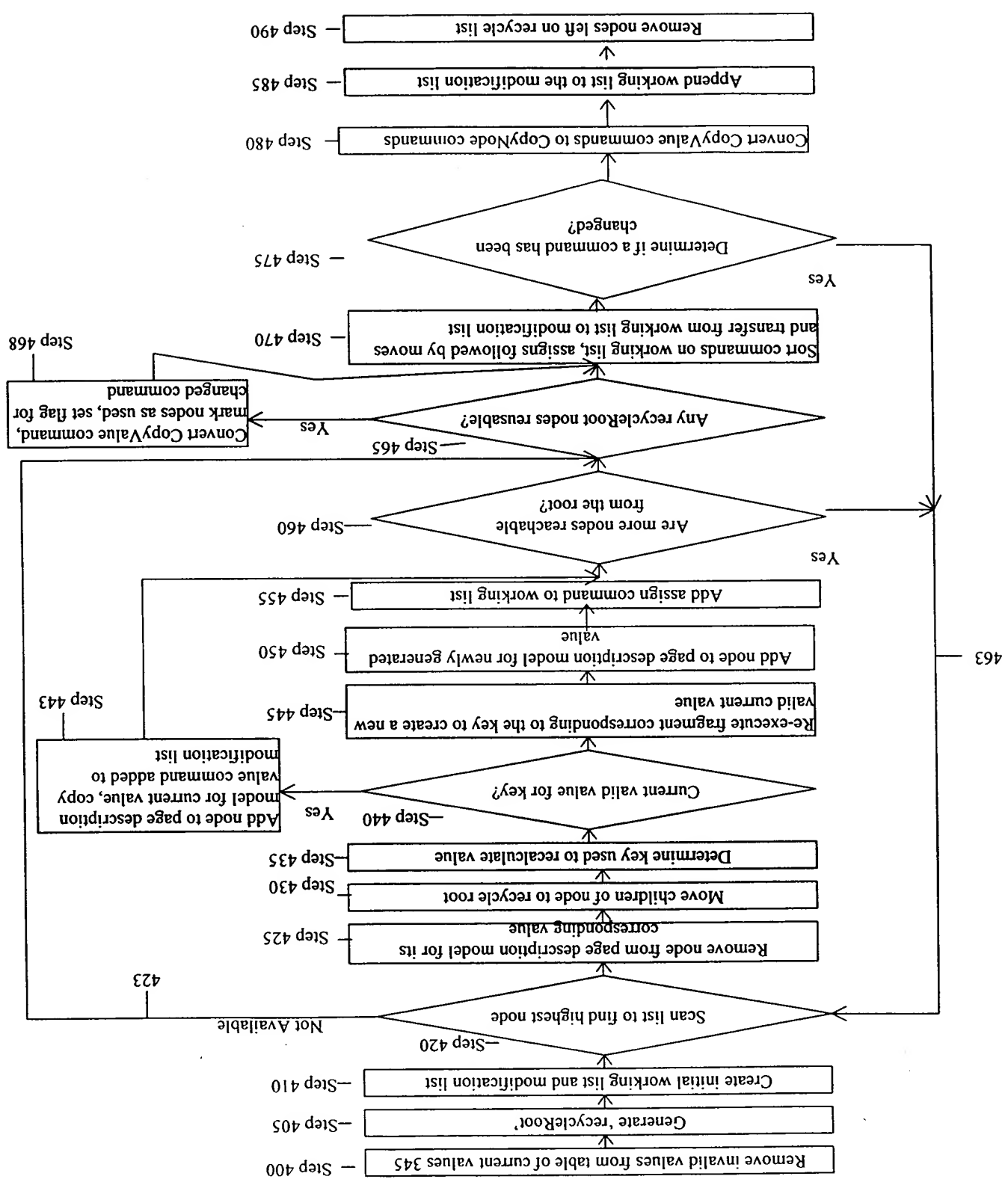
300,



816F

FIG. 9 is a flowchart illustrating a process for managing a list of nodes and commands. The process begins with Step 400, where invalid values are removed from a table of current values 345. This is followed by Step 405, where a 'recycleRoot' is generated, and Step 410, where an initial working list and modification list are created. The process then enters a loop starting with Step 420, where a scan is performed to find the highest node. If the node is 'Not Available' (Step 423), the process proceeds to Step 440, where a current valid value for a key is determined. If the current valid value is for a key (Step 440), the process proceeds to Step 445, where a new valid current value is determined. This is followed by Step 450, where a node is added to the page description model for newly generated value, and Step 455, where the command is added to the working list. The process then proceeds to Step 460, where it checks if more nodes are reachable from the root. If 'Yes' (Step 460), the process proceeds to Step 465, where it checks if any recycleRoot nodes are reusable. If 'Yes' (Step 465), the process proceeds to Step 470, where commands on the working list are sorted and transferred to the modification list. This is followed by Step 475, where it checks if a command has been changed. If 'Yes' (Step 475), the process proceeds to Step 480, where CopyValue commands are converted to CopyNode commands, and Step 485, where the working list is appended to the modification list. Finally, Step 490, where nodes left on the recycle list are removed, and the process loops back to Step 420. If 'No' (Step 460), the process proceeds to Step 463, where it checks if the current valid value is for a key. If 'Yes' (Step 463), the process proceeds to Step 440. If 'No' (Step 463), the process proceeds to Step 420.

FIG. 9



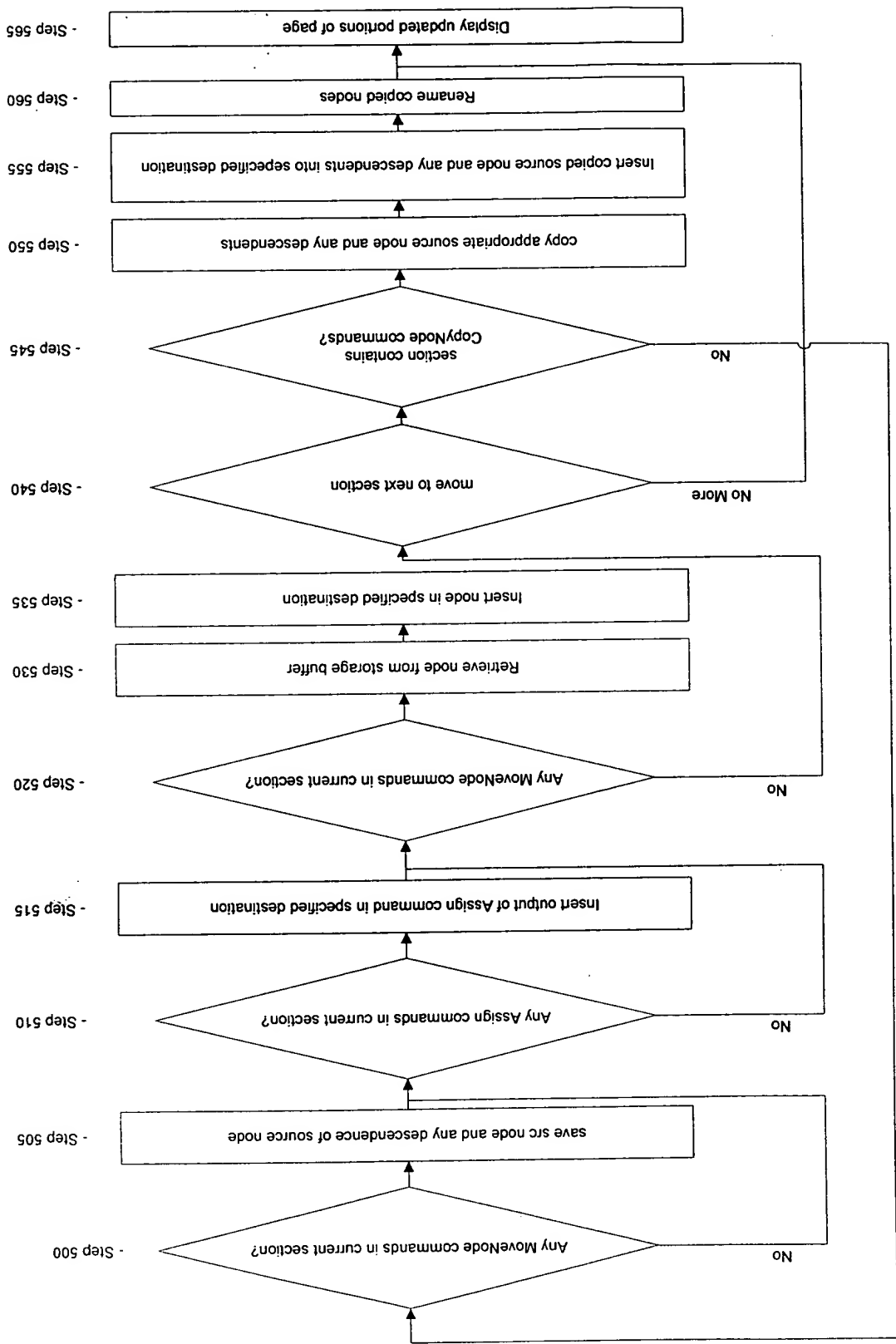


FIG. 10

FIG. 11a

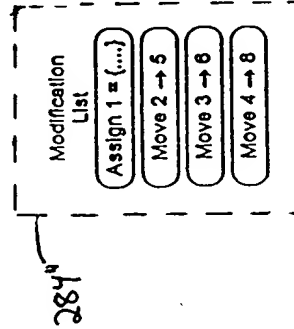
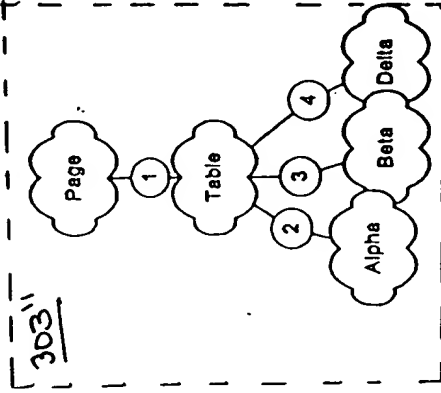


FIG. 11b

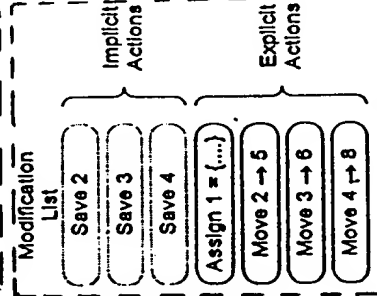
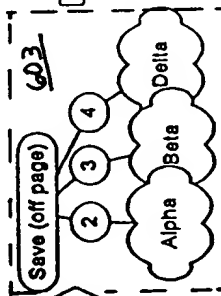
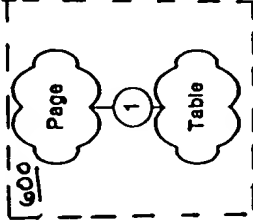


FIG. 11c

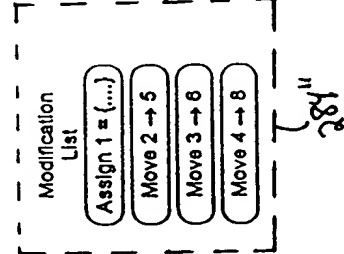
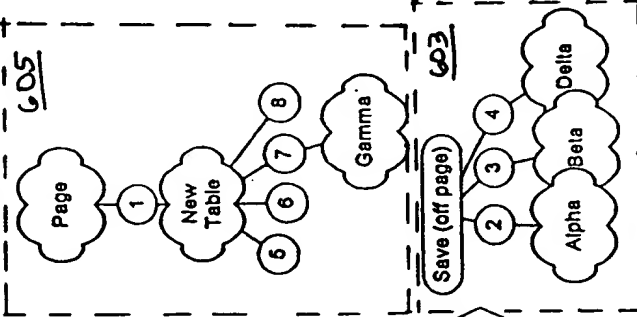


FIG. 11d

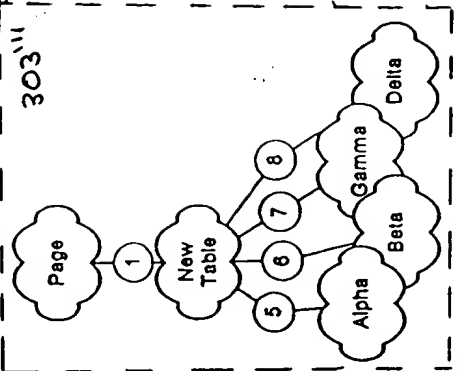


FIG. 11

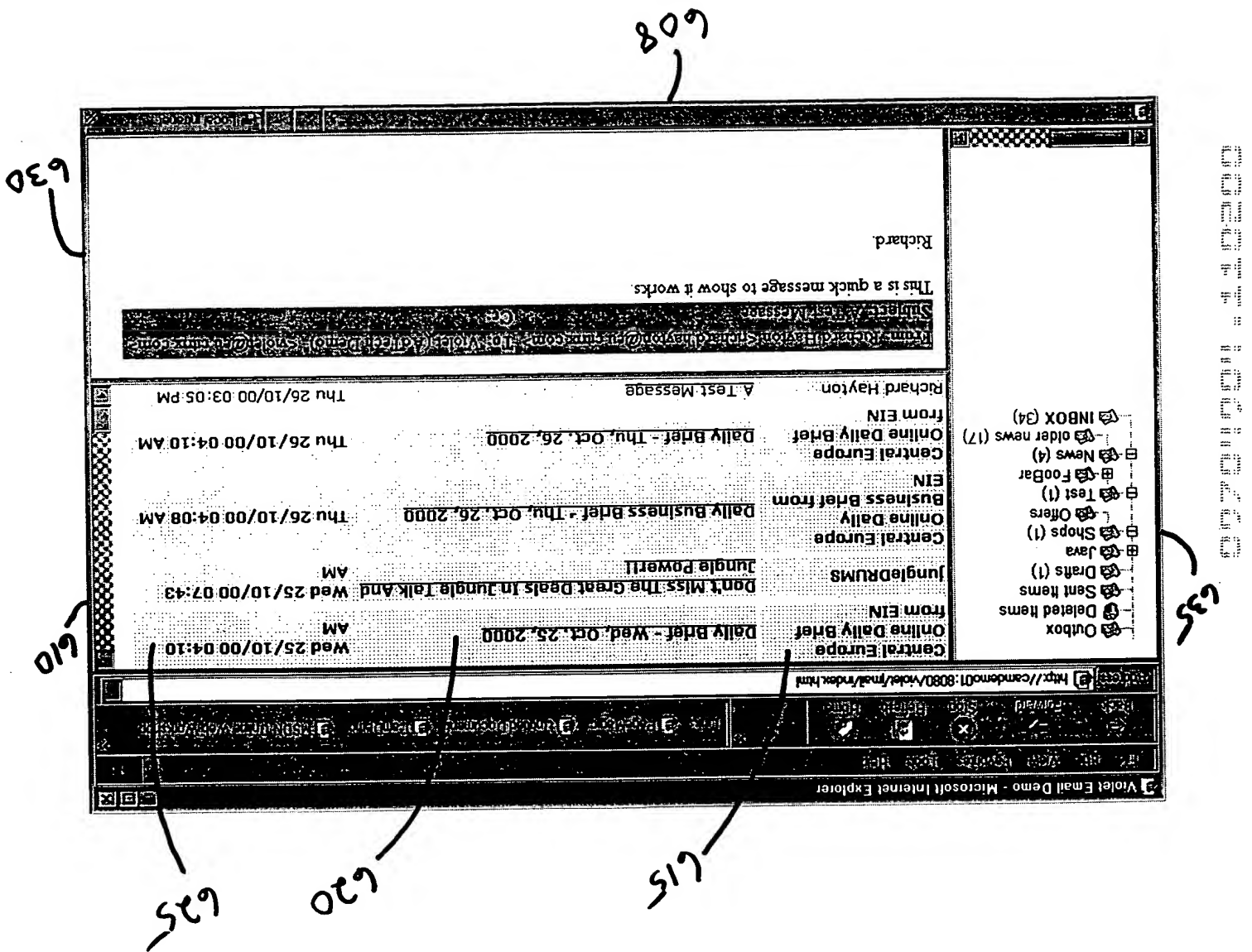


FIG. 12